Removing Mistracking of Multibody Motion Video Database Hopkins155

Yasuyuki Sugaya[1]
sugaya@iim.cs.tut.ac.jp
Yuichi Matsushita[1]
matusita@iim.cs.tut.ac.jp
Kenichi Kanatani[2]
kanatani2013@yahoo.co.jp

[1]Department of Information and Computer Sciences
Toyohashi University of Technology
Toyohashi, Aichi 441-8580 Japan
[2]Department of Computer Science
Okayama University
Okayama 700-8530 Japan

Abstract

Many mathematical techniques have been presented for classifying feature point trajectories over multibody motion video sequences into different motions, and most are applied to the Hopkins155 database for evaluating their performance. In this paper, we point out that Hopkins155 has problems and that correct performance evaluation is not necessarily done using it. We create a new database by removing incorrect trajectories from Hopkins155. The basic principle of mistracking removal is the fact that correct trajectories all belong to parallel 2-D affine spaces in a high-dimensional space if all motions are translational and that parallel 2-D affine spaces are included in a 3-D affine space. Noting that if the image sequence is divided into short intervals, individual motions can be regarded as approximately translational in each interval, we detect incorrect trajectories by repeated plane fitting in the 3-D space by RANSAC. We point out why conventional RANSAC voting does not work and demonstrate that out method allows us to tell in which frames incorrect trajectoires occurred. The performance of multibody motion segmentation can be correctly evaluated using our database.

1 Introduction

Separating independently moving objects in a video stream has attracted attention of many researchers. The most classical work is by Costeira and Kanade [2], who proposed a segmentation algorithm based on the shape interaction matrix. Gear [5] used the reduced row echelon form and graph matching. Ichimura [7] used the discrimination criterion of Otsu [16]. He also used the QR decomposition [8]. Inoue and Urahama [9] introduced fuzzy clustering. Kanatani [11, 12, 13] combined the geometric AIC [10] and robust clustering. Wu et al. [25] introduced orthogonal subspace decomposition. Gruber et al. [6] applied an EM algorithm to the factorization method. Sugaya Kanatani [20, 21] proposed a multistage learning strategy using hierarchical models. Vidal et al. [23, 24] applied GPCA (Generalized Principal Component Analysis), which fits high-degree polynomial to multiple subspaces. Fan et al. [4] and Yan and Pollefeys [26] introduced new voting schemes for classifying oints into different subspaces in high dimensions. Schindler et al. [18] and Rao et al. [17] incorporated model selection based on the MDL (Minimum Description Length) principle. Recently, various schemes are investigated for expressing individual data as linear combinations of a small number of other data and separating the associated similarity graph. These include SSC (Sparse Subspace Clustering) [1, 3], LRR (Low Rank Representation) [14] and LSR (Least Squares Regression) [15].

2 Issues of Existing Methods

All existing methods are based on the fact that, under affine camera modeling, trajectories of image points in the same motion belong to a common 4-D subspace or 3-D affine space in a high-dimensional space. Hence, typical situations where correct segmentation cannot be done are:

1. Imaging geometry cannot be modeled by an affine camera due to foreshortening effects.
2. Degeneracies occurs in the trajectory space, depending on motion types.
3. Feature points are not extracted in the correct positions in individual frames.
4. Different feature points are matched between different frames, resulting in incorrect trajectories.

However, most of the existing methods only evaluate their performance in terms of the correct segmentation ratio, and the origin of incorrect segmentation have rarely been investigated. This tendency has became conspicuous since Tron and Vidal [22] presented the Hopkins155 video database. This hides the details of the working of algorithms. Rather than proposing new mathematical techniques and testing them on Hopkins155, we need to scrutinize individual cases as to how and why correct segmentation is done or not done.

According to our experiences, the above item 1 (foreshortening) need not be worried about, because in situations when motions of multiple objects are captured by a video camera over a certain length of time, the camera is usually far apart from the objects. In contrast, item 2 (degeneracy) is an important issue, because almost all "natural" motions lead to degenerate or nearly degenerate trajectories in the trajectory space, typical instances including the objects simply translating in the image. This was extensively studied by Sugaya and Kanatani [20, 21], who proposed a multi-stage learning strategy, assuming various degenerate motion models and starting from special to general models.

On the other hand, items 3 and 4 do not seem to have been fully investigated. In the past, any trajectoires that do not belong to a 4-D subspace or a 3-D affine space are collectively called "outliers", and outlier removal procedures have been proposed using RANSAC in the trajectory space [19]. In this paper, we point out that such across-the-board voting is not effective and why. In the past, however, performance has often tested using non-realistic data generated such as adding Gaussian noise to the feature point locations or introducing artificial points sampled from a uniform distribution. For realistic evaluation, we need to pay more attention to the data data generating mechanism.

3 Scrutiny of Hopkins155

In this paper, we consider the Hopkins155 database[1], which most researchers use for performance evaluation of multibody motion segmentation algorithms. The video images it provides are carefully prepared so that all the trajectoires appear to be correct. We have found, however, that many subtle problems exist and that they are very difficult to detect by visual inspection. So, we build a supporting interface.

The basic principle is as follows. A trajectory of a feature point over M frames is represented by a point in a $2M$-D space, and trajectories of points in a common rigid motion are constrained to be in a 3-D affine space in $2M$-D under affine camera modeling [11, 12, 13]. However, many real motions are translations and rotations within the image frame. In this case, the 3-D affine space degenerate into a 2-D affine space. Furthermore, if multiple motions are all in-plane translations, the

[1] http://www.vision.jhn.edu/data/hopkins155

corresponding 2-D affine spaces are all parallel to each other, and there exist a 3-D affine space that include all these 2-D affine spaces. Hence, if the $2M$-D trajectory space is projected onto that 3-D affine space, we can visualize all the trajectories as points in 3-D. Even if the motions are not exactly translational but nearly so, which is the case in most natural scenes, we can view all the trajectory as points on nearly parallel curved surfaces in 3-D, and we can easily find incorrect trajectories, because they stick out from the others on a common surface. We take a step further and automate this process, providing a reliability measure that tells to what extent each trajectory is likely to be correct.

4 Affine Camera Modeling

We first summarize the well known fact that the trajectories of points in a common rigid motion are constrained in a 3-D affine space in the $2M$-D trajectory space. Suppose we track N feature points $\{p_\alpha\}$ over M frames. Let $(x_{\kappa\alpha}, y_{\kappa\alpha}), \kappa = 1, ..., M, \alpha = 1, ..., N$, be the image coordinates of the αth point p_α in the κth frame. Its motion history is represented by the $2M$-D vector

$$\mathbf{p}_\alpha = (x_{1\alpha}, y_{1\alpha}, x_{2\alpha}, y_{2\alpha}, ..., x_{M\alpha}, y_{M\alpha})^\top, \tag{1}$$

which we simply call the "trajectory" of p_α. Thus, all trajectories can be identified with points in an $2M$-D space.

We regard the camera as fixed, relative to which the scene and multiple objects are moving. Take an XYZ coordinate system fixed to the camera with the Z axis in the optical axis direction, and regard it as the world coordinate system. Define a coordinate system attached to a moving object, and let $(a_\alpha, b_\alpha, c_\alpha)$ be the coordinates of p_α with respect to that object coordinate system. If $\mathbf{t}_\kappa$ and $\{\mathbf{i}_\kappa, \mathbf{j}_\kappa, \mathbf{k}_\kappa\}$ are, respectively, the origin and the basis vectors of the object coordinate system described with respect to the world coordinate system (Fig. 1(a)), the point p_α in the κth frame is in the following position with respect to the world coordinate system:

$$\mathbf{r}_{\kappa\alpha} = \mathbf{t}_\kappa + a_\alpha \mathbf{i}_\kappa + b_\alpha \mathbf{j}_\kappa + c_\alpha \mathbf{k}_\kappa. \tag{2}$$

Under the affine camera modeling, which generalizes orthographic, weak perspective, and paraperspective projections, the point $\mathbf{r}_{\kappa\alpha}$ in Eq. (2) is projected onto a point $(x_{\kappa\alpha}, y_{\kappa\alpha})$ in the image as follows:

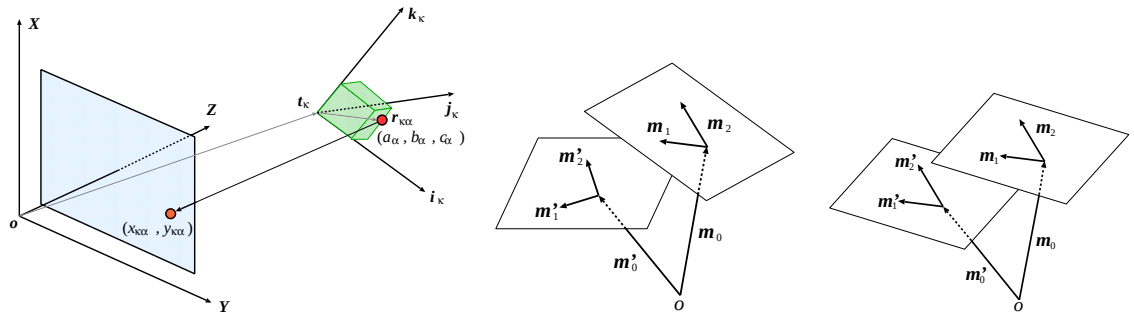
$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \mathbf{A}_\kappa \mathbf{r}_{\kappa\alpha} + \mathbf{b}_\kappa. \tag{3}$$

Here, $\mathbf{A}_\kappa$ and $\mathbf{b}_\kappa$ are, respectively, 2×3 matrix and a 2-D vector determined by the position and orientation of the object coordinate system and the camera parameters for the κth frame. If Eq. (2) is substituted, Eq. (3) becomes

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \tilde{\mathbf{m}}_{0\kappa} + a_\alpha \tilde{\mathbf{m}}_{1\kappa} + b_\alpha \tilde{\mathbf{m}}_{2\kappa} + c_\alpha \tilde{\mathbf{m}}_{3\kappa}, \tag{4}$$

where $\tilde{\mathbf{m}}_{0\kappa}$, $\tilde{\mathbf{m}}_{1\kappa}$, $\tilde{\mathbf{m}}_{2\kappa}$, and $\tilde{\mathbf{m}}_{3\kappa}$ are 2-D vectors determined by the position and orientation of the object coordinate system and the camera parameters for the κth frame. If we vertically align all such 2-D vectors for $\kappa = 1, ..., M$ into a $2M$-D vector, the trajectory $\mathbf{p}_\alpha$ in Eq. (1) has the following expression:

$$\mathbf{p}_\alpha = \mathbf{m}_0 + a_\alpha \mathbf{m}_1 + b_\alpha \mathbf{m}_2 + c_\alpha \mathbf{m}_3. \tag{5}$$



(a) (b) (c)

Figure 1: (a) Affine camera modeling. (b) The trajectories of points in in-plane motions are constrained to be in 2-D affine spaces. (c) The trajectories of translating points are constrained to be in parallel 2-D affine spaces.

Here, $\mathbf{m}_i$, i = 0, ..., 3, are the $2M$-D vectors obtained by vertically aligning $\tilde{\mathbf{m}}_{i\kappa}$, κ = 1, ..., M, for all the frames.

Equation (5) shows that the trajectories $\mathbf{p}_\alpha$ of points in a common rigid motion are constrained to be in the 4-D subspace of the $2M$-D trajectory space spanned by $\{\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$. However, the coefficient of $\mathbf{m}_0$ is 1 irrespective of α. Hence, $\mathbf{p}_\alpha$ is constrained to be in the 3-D affine space passing through $\mathbf{m}_0$ and spanned by $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$.

5 Visualization of Trajectories

In many scenes, objects (including the background) simply translate and rotate within the image plane. If the object coordinate basis vector $\mathbf{k}_\alpha$ to taken to be in the Z-axis (= the camera optical axis), the vector $\mathbf{m}_3$ in Eq. (5) is $\mathbf{0}$ for such in-plane motions under the affine camera modeling (recall that the object coordinate system can be arbitrarily defined as long as it is fixed to the object). Hence, all the trajectories of points of that object are constrained to be the 2-D affine space passing through $\mathbf{m}_0$ and spanned by $\{\mathbf{m}_1, \mathbf{m}_2\}$ (Fig. 1(b)).

If furthermore all objects are simply translating without rotation, the basis vectors $\mathbf{i}_\kappa$ and $\mathbf{j}_\kappa$ in Eq. (2) can be aligned to the basis vectors $\mathbf{i}$ and $\mathbf{j}$ of the world coordinate system. Since these are common to all the motions, the vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ in Eq. (5) are also common to all the motions. Hence, the 2-D affines spaces of the motions are *parallel* to each other (Fig. 1(c)). It follows that there exists a 3-D affine space that include these parallel 2-D affine spaces. Hence, if all the trajectories in the $2M$-D space are projected onto that 3-D affine space, we can visualize the points lying on parallel planes in 3-D. Even if the motions are not exactly translational but nearly translational, which is the case in most natural scenes, we can see points on nearly parallel curved surfaces in 3-D. The actual procedure is done by principal component analysis as follows:

1. Compute the centroid $\mathbf{p}_C$ of the trajectories $\mathbf{p}_\alpha$, α = 1, ..., N and the deviations $\tilde{\mathbf{p}}_\alpha$ from $\mathbf{p}_C$:

$$\mathbf{p}_C = \frac{1}{N}\sum_{\alpha=1}^{N}\mathbf{p}_\alpha, \quad \tilde{\mathbf{p}}_\alpha = \mathbf{p}_\alpha - \mathbf{p}_C. \tag{6}$$

2. Compute the singular value decomposition of the $2M \times N$ matrix

$$(\tilde{\mathbf{p}}_1, ..., \tilde{\mathbf{p}}_N) = \mathbf{U}\mathrm{diag}(\sigma_1, ..., \sigma_r)\mathbf{V}^\top, \tag{7}$$

where $r = \min(2M, N)$, $\mathbf{U}$ is a $2M \times r$ matrix with r orthonormal columns, $\mathbf{V}$ is an $N \times r$ matrix with r orthonormal columns, and $\sigma_1 \geq \cdots \geq \sigma_r$ (≥ 0) are the singular values.

3. Let $\mathbf{u}_i$ be the ith column of $\mathbf{U}$, and compute the following 3-D vectors $\mathbf{r}_\alpha$, α = 1, ..., N:

$$\mathbf{r}_\alpha = ((\tilde{\mathbf{p}}_\alpha, \mathbf{u}_1), (\tilde{\mathbf{p}}_\alpha, \mathbf{u}_2), (\tilde{\mathbf{p}}_\alpha, \mathbf{u}_3))^\top, \tag{8}$$

where and hereafter we denote the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ by $(\mathbf{a}, \mathbf{b})$.

Geometrically, we are translating the coordinate system of the $2M$-D trajectory space so that the origin is at the centroid $\mathbf{p}_C$, computing the three vectors (the columns of $\mathbf{U}$) that span the affine space, and expressing all the trajectories as their linear combinations.

6 Detection of Incorrect Trajectoires

Using the above technique, we can visualize all correct trajectories as points in 3-D on nearly parallel curved surfaces in 3-D, and we can easily find incorrect trajectories, because they stick out from the others. Such irregular points are very easily discernible if the 3-D space is displayed on a display by continuously moving the viewpoint. If we re-examine such irregular trajectories on the original video images, we can find tacking errors that are often overlooked on the first visual inspection and also tell where and how they occur.

This process is very effective but involves human intervention. We next automate this. The core idea is the fact that even if the object and background motions are not exactly translational, they are nearly translational if the video stream is divided into very short sequences. Hence, incorrect trajectories can be detected by fitting multiple planar surfaces in 3-D, and RANSAC is suited for this purpose. The reliability of individual trajectories can be evaluated by the distance to the nearest fitted plane. Finally, we obtain a reliability measure of each trajectory by integrating its behavior in all the intervals. We now describe this procedure step by step.

We fit a plane

$$Ax + By + Cz + Df_0 = 0 \tag{9}$$

to points $(x_\alpha, y_\alpha, z_\alpha)$, α = 1, ..., N, in 3-D, where f_0 is a scale normalization constant to stabilize numerical computation. If we define 3-D vectors $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ by

$$\boldsymbol{\xi} = (x, y, z, f_0)^\top, \quad \boldsymbol{\theta} = (A, B, C, D)^\top, \tag{10}$$

Eq. (9) is written as $(\boldsymbol{\xi}, \boldsymbol{\theta}) = 0$. In order to remove the scale indeterminacy, we normalize $\boldsymbol{\theta}$ to unit norm: $\|\boldsymbol{\theta}\| = 1$. The simplest fitting scheme is least squares (LS). Letting $\boldsymbol{\xi}_\alpha = (x_\alpha, y_\alpha, z_\alpha, f_0)^\top$, we minimize

$$J = \sum_{\alpha=1}^{N} (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 = (\boldsymbol{\theta}, \underbrace{\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}_{\equiv \mathbf{M}} \boldsymbol{\theta}) = (\boldsymbol{\theta}, \mathbf{M}\boldsymbol{\theta}). \tag{11}$$

The solution is the unit eigenvector of the matrix $\mathbf{M}$ for the smallest eigenvalue. We fit multiple planes to multiple points in 3-D by the following RANSAC:

1. Randomly choose three from among points $\mathbf{r}_\alpha$, α = 1, ..., N.
2. Fit a plane to the selected three points and compute $\boldsymbol{\theta}$ by LS.

Figure 2: Three video sequences in Hopkins155. The marks □ and × indicate feature point locations regarded as correctly tracked and incorrectly tracked, respectively, by our procedure.

3. Let S be the number of points $\mathbf{r}_\alpha$ that satisfy

$$\frac{(\mathbf{r}_\alpha, \boldsymbol{\theta})^2}{\theta_1^2 + \theta_2^2 + \theta_3^2} \leq \sigma^2, \tag{12}$$

   where θ_i is the ith component of $\boldsymbol{\theta}$. The left hand side is the square distance of point $\mathbf{r}_\alpha$ from the fitted plane, and σ is the standard deviation of feature point detection accuracy, which is empirically set.
4. Repeat the above computation many times and find the value $\boldsymbol{\theta}$ that maximize S.
5. Remove those points $\mathbf{r}_\alpha$ that satisfy

$$\frac{(\mathbf{r}_\alpha, \boldsymbol{\theta})^2}{\theta_1^2 + \theta_2^2 + \theta_3^2} < \sigma^2 \chi^2_{1;99}, \tag{13}$$

   where $\chi^2_{r;a}$ is the ath percentile of χ^2 distribution with r degrees of freedom. This means that we retain those points that cannot be regarded as deviated from the fitted plane by Gaussian noise of mean 0 and standard deviation σ with 1% significance level.

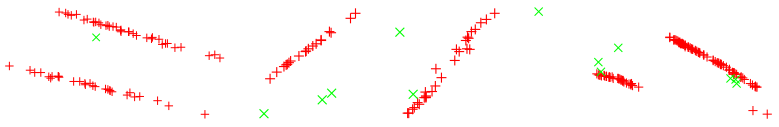
We integrate the result in all intervals by defining the reliability index for the ith interval using the sigmoid function as follows:

$$P(\mathbf{p}_\alpha^{(i)}) = \frac{1}{1 + e^{-(d_\alpha^{(i)} - \sigma^2 \chi^2_{1;99})}}. \tag{14}$$

Here, $\mathbf{p}_\alpha^{(i)}$ is the vector that describe the partial trajectory of $\mathbf{p}_\alpha$ over the ith interval, and $d_\alpha^{(i)}$ is the left-hand side of Eq. (13) for the ith interval. However, we regard those points that satisfy Eq. (13) as correct and let $P(\mathbf{p}_\alpha^{(i)}) = 0$. We integrate the results in all the interval in the following form (the trajectory is more likely to be incorrect if it is larger):

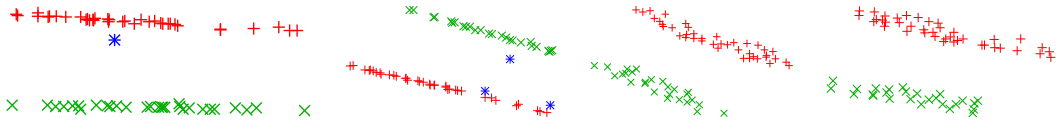
$$L(\mathbf{p}_\alpha) = \Pi^K_{i|P(\mathbf{p}_\alpha^{(i)}) \neq 0} P(\mathbf{p}_\alpha^{(i)}). \tag{15}$$

Here, K is the number of intervals.



(a) (b) (c)

Figure 3: 3-D visualization tracking trajectory in the first five frames in the sequences in Fig. 2.



1st interval 2nd interval 3rd interval 4th interval

Figure 4: 3-D visualization of partial trajectories in four consecutive five-frame intervals of Fig. 2(a).

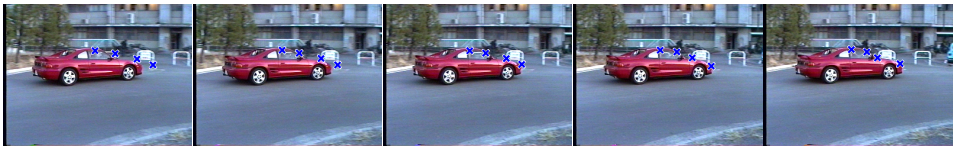
7 Experiments

Figure 2 shows three videos of Hopkins155. We added new feature tracking and divided the sequence into five-frame intervals with one frame overlaps. Five decimated frames are shown in Fig. 2. The marks $\square$ and $\times$ indicate feature point locations regarded as correctly tracked and incorrectly tracked, respectively. We set σ =1.0, which we judged to be reasonable according to our experiences. Figure 3 shows 3-D visualization of partial trajectories in the first five-frame intervals. We can clearly see that correct trajectories lie on two planes and that incorrect trajectories stick out from them.

Next, we closely examined each five-frame interval of the sequence of Fig. 2(a). Figure 4 shows 3-D visualization of partial trajectories in each interval. The marks $+$ indicted trajectories regarded as on a plane in the first RANSAC run, the marks $\times$ as on another plane in the second run, and the mark $*$ indicate trajectories regarded as incorrect. Then, we re-examined the corresponding video frames to see how and where incorrect tracking occurred. We find that one feature point of a moving vehicle (the leftmost one in Fig. 5) is wrong in the 5th frame. However, this point, although in a wrong position, is tracked correctly in the subsequent frames. Hence, the trajectory of this point is not judged to be incorrect. On the other hand, the trajectories judged to be incorrect from the 6th through the 10th frames (the three from right in Fig. 5) should belong to the background but are occluded by the moving vehicle and tracked as feature points of the vehicle. Thus, their trajectories are regarded as correct background points in the first interval, then regarded as incorrect in the second interval, and subsequently regarded as correct as vehicle points.

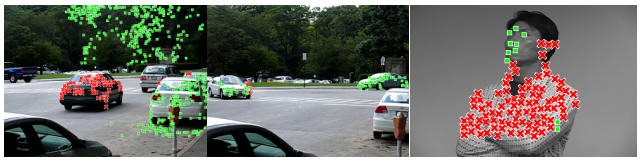
This shows how difficult it is to tell whether the feature tracking is correct or not simply by seeing individual frames. Our procedure can not only find such incorrect tracking automatically but also tell where the errors have occurred. As a comparison, we conducted the standard outlier procedure of fitting 4-D subspaces by RANSAC in the $2M$-D trajectory space and found that 50% of our detected errors were not detected. If 3-D affine spaces are fitted by RANSAC, again 50% of our detected errors were not detected. Thus, the RANSAC in $2M$-D is not effective, and voting within in the reduced 3-D space is indispensable.

Among the natural scenes of Hopkins155 (we do not consider artificial images created by CG), there are 35 scenes consisting of two motions i.e., a moving object and a moving background. We applied the algorithm of Sugaya and Kanatani [21] and found that all scenes were successfully segmented but three. The three unsuccessful scenes are shown in Fig. 6 along with the correctness ratio (above). After removing incorrect trajectories using our technique, we applied the same algorithm



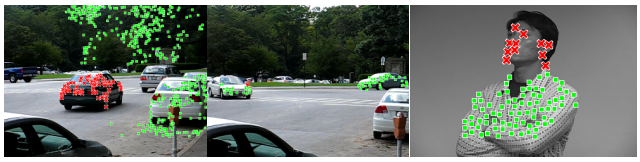
4th frame 5th frame 6th frame 7th frame 8th frame

Figure 5: Incorrect feature point tracking in Fig. 2(a).



(a) cars2 (b) cars5_g12 (c) head

99.38% 61.9% 91.92%



100% 61.7% 100%

Figure 6: Segmentation results and the correctness ratios by the algorithm of Sugaya and Kanatani [21]. Above: original Hopkins155. Below: after incorrect trajectories have been removed. [21]

again. The result is shown below. We can see that 100% correctness is achieved for the scenes (a) and (c). We conclude that for these two scenes there were no problems in the segmentation algorithm and that the low performance was due to the database.

For the scene (b), on the other hand, something should be wrong with the algorithm. We found that in the program the initial segmentation was poor and not much improved by subsequent EM iterations. Since the performance of EM algorithms heavily depends on the initial value, this suggests that the performance could be made higher by improving the initial segmentation, for which the algorithm of Sugaya and Kanatani [21] used the GPCA (Generalized Principal Component Analysis) of Vidal et al. [23, 24] for fitting two planes simultaneously. We tentatively replaced it with the progressive plane fitting by RANSAC described in Sec. 6 and found that 100% correctness was achieved. Thus, in this case the problem was not in the database but in the algorithm, and this observation successfully lead to the improvement of the algorithm.

In the past, new mathematical techniques have been proposed one after another and tested on Hopkins155 for performance evaluation, and the correctness ratio has been regarded as a property of that algorithm, not of the data. However, we need to know if the apparent high or low performance is due to the data or due to the algorithm. Without this knowledge, we cannot improve the algorithm. For correct evaluation, we need a reliable database. For this purpose, we created a new database[2] by removing incorrect trajectories using our technique from the above mentioned 35 scenes of Hopkins155.

[2]http://www.iim.cs.tut.ac.jp/T-Hopkins/

8 Concluding Remarks

We presented a powerful method for detecting incorrect tracking trajectories in multibody motion video sequences. The basic principle is the fact if image motions are translational, correct trajectories belong to parallel 2-D affine spaces included in a 3-D affine space in the trajectory space. We automated this process by noting that if the image sequence is divided into short intervals, image motions can be regarded as approximately translational in each interval. We detected incorrect trajectories by repeated plane fitting in the 3-D space by RANSAC and demonstrated that out method can not only detect tracking errors that are easily overlooked by visual inspection but also tell use where they occurred. We also pointed out why outliner removal based on RANSAC in $2M$-D is not effective. Using our method, we removed incorrect trajectories from Hopkins155 and create a new database, which is expected to serve as an indispensable benchmark for multibody motion segmentation study.

Acknowledgments: This work was supported in part by JSPS Grant-in-Aid for Young Scientists (B 23700202) and for Challenging Exploratory Research (24650086).

References

[1] B. Cheng, J. Yang, S. Yan, Y. Fu and T. S. Huang. Learning with l^1-graph for image analysis, *IEEE Trans Patt. Anal. Mach. Intell.*, 19(4), 858–866, 2010.

[2] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *Int. J. Comput. Vis.*, 29(3): 159–179, 1998.

[3] E. Elhamifar and R. Vidal. Sparse subspace clustering, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Miami FL, U.S.A., June 2009, pp. 2790–2797.

[4] Z. Fan, J. Zhou and Y. Wu. Multibody grouping by inference of multiple subspace from high-dimensional data using oriented-frames. *IEEE Trans Patt. Anal. Mach. Intell.*, 28(1): 91–105, 2006.

[5] C. W. Gear. Multibody grouping from motion images. *Int. J. Comput. Vision*, 29(2): 133–150, 1998.

[6] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*, Washington, DC, U.S.A., June/July 2004, Vol. 1, pp. 769–775.

[7] N. Ichimura. Motion segmentation based on factorization method and discriminant criterion. *Proc. 7th Int. Conf. Comput. Vis.*, September 1999, Kerkyra, Greece, Vol. 1, pp. 600–605.

[8] N. Ichimura. Motion segmentation using feature selection and subspace method based on shape space,. *Proc. 15th Int. Conf. Pattern Recog.*, September 2000, Barcelona, Spain, Vol. 3, pp. 858–864.

[9] K. Inoue and K. Urahama. Separation of multiple objects in motion images by clustering. *Proc. 8th Int. Conf. Comput. Vis.*, July 2001, Vancouver, Canada, Vol. 1, pp. 219–224.

[10] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, The Netherlands, 1996; Reprinted, Dover, New York, NY, U.S.A., 2005.

[11] K. Kanatani. Motion segmentation by subspace separation and model selection. *Proc. 8th Int. Conf. Comput. Vis.*, Vancouver, Canada, July 2001, Vol. 2, pp. 301–306.

[12] K. Kanatani. Evaluation and selection of models for motion segmentation. *Proc. 7th Euro. Conf. Comput. Vis.*, Copenhagen, Denmark, June 2002, Vol. 3, pp. 335–349.

[13] K. Kanatani. Motion segmentation by subspace separation: Model selection and reliability evaluation. *Int. J. Image Graphics*, 2(2): 179–197, 2004.

[14] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu and Y. Ma. Robust recovery of subspace structure by low-rank representation, *IEEE Trans Patt. Anal. Mach. Intell.*, 35(1): 171–184, 2013.

[15] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang and S. Yan. Robust and efficient subspace segmentation via least squares regression, *Proc. Euro. Conf. Comput. Vision.*, October 2011, Firenze, Italy, Vol. 7, pp. 347–360.

[16] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.*, 9(1): 62–66, 1979.

[17] S. R. Rao, R. Tron, R. Viadl and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*, June 2008, Anchorage, AK, U.S.A.

[18] K. Schindler, D. Suter and H. Wang. A model-selection framework for multibody structure-and-motion of image sequences. *Int. J. Comput. Vision*, 79(2): 159–177, 2008.

[19] Y. Sugaya and K. Kanatani. Outlier removal for motion tracking by subspace separation. *IEICE Trans. Inf. Syst.*, E86-D(6): 1095–1102, 2003.

[20] Y. Sugaya and K. Kanatani. Multi-stage optimization for multi-body motion segmentation. *IEICE Trans. Inf. Syst.*, E87-D(7): 1935–1942, 2004.

[21] Y. Sugaya and K. Kanatani. Improved multistage learning for multibody segmentation. *Proc. 5th Int. Conf. Comput. Vis. Theory Appl./*, May 2010, Angers, France, Vol. 1, pp. 199–206.

[22] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*, June 2007, Minneapolis, MN, U.S.A.

[23] R. Vidal, Y. Ma and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Trans. Patt. Anal. Mach. Intell.*, 27(12): 1945–1959, 2005.

[24] R. Vidal, R. Tron and R. Hartley. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *Int. J. Comput. Vision*, 79(1): 85–105, 2008.

[25] Y. Wu, Z. Zhang, T. S. Huang and J. Y. Lin. Multibody grouping via orthogonal subspace decomposition, sequences under affine projection. *Proc. IEEE Conf. Comput. Visi. Pattern Recog.*, December 2001, Kauai, HI, U.S.A. Vol. 2, pp. 695–701.

[26] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulate, rigid, non-rigid, degenerate and nondegenerate. *Proc. Euro. Conf. Comput. Vision.*, May 2006, Graz, Austria, Vol. 4, pp. 94–104.